REMARKS

Claims 1 - 20 are pending in the present Application. In the above-identified Office Action, the Examiner objected to the DRAWINGS. Claims 1 - 20 were rejected under 35 U.S.C. §102(a) as being anticipated by Vahalia et al.

In reviewing the Specification, Applicants have encountered a few typographical/grammatical errors which have been corrected.

Applicants have also amended the figures. A set of corrected figures is provided with RED markings. The Red markings represent changes that are made to the figures. In addition, a new set of figures, which incorporates the changes made to the figures, is also provided. The new set of figures includes an added figure (a new Fig. 8). No new matter is added to the Application as a result to this new figure since description of the figure can be found on page 15, lines 13 - 20. Original Fig. 8 has been relabeled Fig. 9. All reference numerals have been appropriately updated.

Further, Applicants have amended independent Claims 1, 6, 11 and 16 to better claim the invention.

For the reasons stated more fully below, Applicants submit that the claims are allowable over the applied reference. Hence, reconsideration, allowance and passage to issue are respectfully requested.

As stated in the SPECIFICATION, some operating systems (OSs) provide a "mount" operation that makes all file systems appear as a single tree, while others maintain a multiplicity of file systems. To mount a file system is to

make the file system available for use at a specified location, the mount point.

Most computer systems, especially those running Microsoft OSs, generally mount all file systems on startup. However, Unix-based computer systems typically do not do so. They only mount certain fie systems on startup. The file systems that are mounted on startup are the ones that contain files that are critical for the OS to function properly. The other file systems are mounted only when needed.

One particular time one of the other file systems is mounted is just before the file system is exported. As mentioned earlier, to export a file system is to make the file system available for NFS clients to mount on their own file systems. When exporting a file system, the mount point as well as the name of the storage device containing the file system must be provided. If the file system is mounted, all the needed information is known; hence, the reason why file systems are mounted before they are exported.

As mentioned before, most Unix-based servers mount some file systems only when they are needed. If a mounted non-critical OS file system has not been used within a pre-defined amount of time, it will be unmounted. This allows for other file systems to be mounted when and if needed. As will be explained later, mounting file systems can be a relatively time-consuming and CPU-intensive endeavor. Thus, mounting file systems only for export purposes may be a great waste of time and energy, especially if the file

systems are subsequently unmounted without ever having been used.

The present invention obviates the need to mount a file system for export purposes only. In accordance with the teachings of the invention, each mount point in a file system has an associated extended attribute file. The extended attribute file contains all information needed to export the file systems that are to be mounted at that mount point. Thus, when a file system is mounted, as are, for example, all OS critical file systems, each extended attribute file associated with a mount point of the mounted file system is consulted to obtain all information needed to export any file that is to be mounted at that point. Once the information is obtained, the file systems may be exported. This way, the file systems themselves need not be mounted in order to obtain the needed information.

The invention is set forth in claims of varying scopes of which Claim 1 is illustrative.

> 1. A method of exporting file systems comprising the steps of:
> ***consulting a file associated with a mount point of a mounted file system to retrieve needed information to export the file systems***, the mount point being the point at which the file systems are mounted on a computer system; and
> exporting the file systems.
> (Emphasis added.)

The Examiner rejected the claims under 35 U.S.C. §102(a) as being anticipated by Vahalia et al., Applicants respectfully disagree.

Vahalia et al. purport to teach a method of recovery from failure of a data processor in a network file server. The network file server includes a first set of data processors for receiving requests from clients, and a second set of data processors for accessing read-write file systems. Each file system is assigned to a data processor in the second set. The data processor to which the file system is assigned has exclusive management of locks on the file system. The file server can detect failure of a failed data processor and automatically recover from the failure. When a failure of a data processor in the first set is detected, a spare data processor is programmed with the logical and physical network addresses of the failed data processor so that the spare data processor assumes the network identity of the failed data processor. When a failure of a data processor in the second set is detected, responsibility for management of the locks on each file system managed by the failed data processor is transferred to an operational data processor in the set.

Each of the data processors in the second set has a directory of the file systems and a database of the mount points for the file systems and the data processor to which each file system is assigned. When a data processor receives a request to access a read-only file, it services the request whether or not the file system being accessed is one of its file systems. If, on the other hand, the request is for a read/write file, it consults the database to determine whether it owns the file system in which the file resides. If it owns the file system, then it services the request. Otherwise, it forwards the request to the

data processor that owns the file system. If the file for which access is requested is in a remote file system, then the data processor exports the request to the proper data processor. Data processors generally know whether a file is in a remote file system if a mount point is reached during file look-up.

However, Vahalia et al. do not teach, show or suggest ***consulting a file associated with a mount point of a mounted file system to retrieve information needed to export file systems*** that are to be mounted at that mount point as claimed.

Therefore, Applicants submit that Claim 1 and its dependent claims should be allowable. Independent Claims 6, 11 and 16 and their respective dependent claims, which all incorporate the above-emboldened-italicized limitations shown in the reproduced Claim 1 above, should be allowable as well.

Therefore, Applicants once more respectfully request reconsideration, allowance and passage to issue of the claims in the application.

Respectfully submitted,
Brown et al.

By: _____
Volel Emile
Attorney for Applicants
Registration No. 39,969
(512) 306-7969

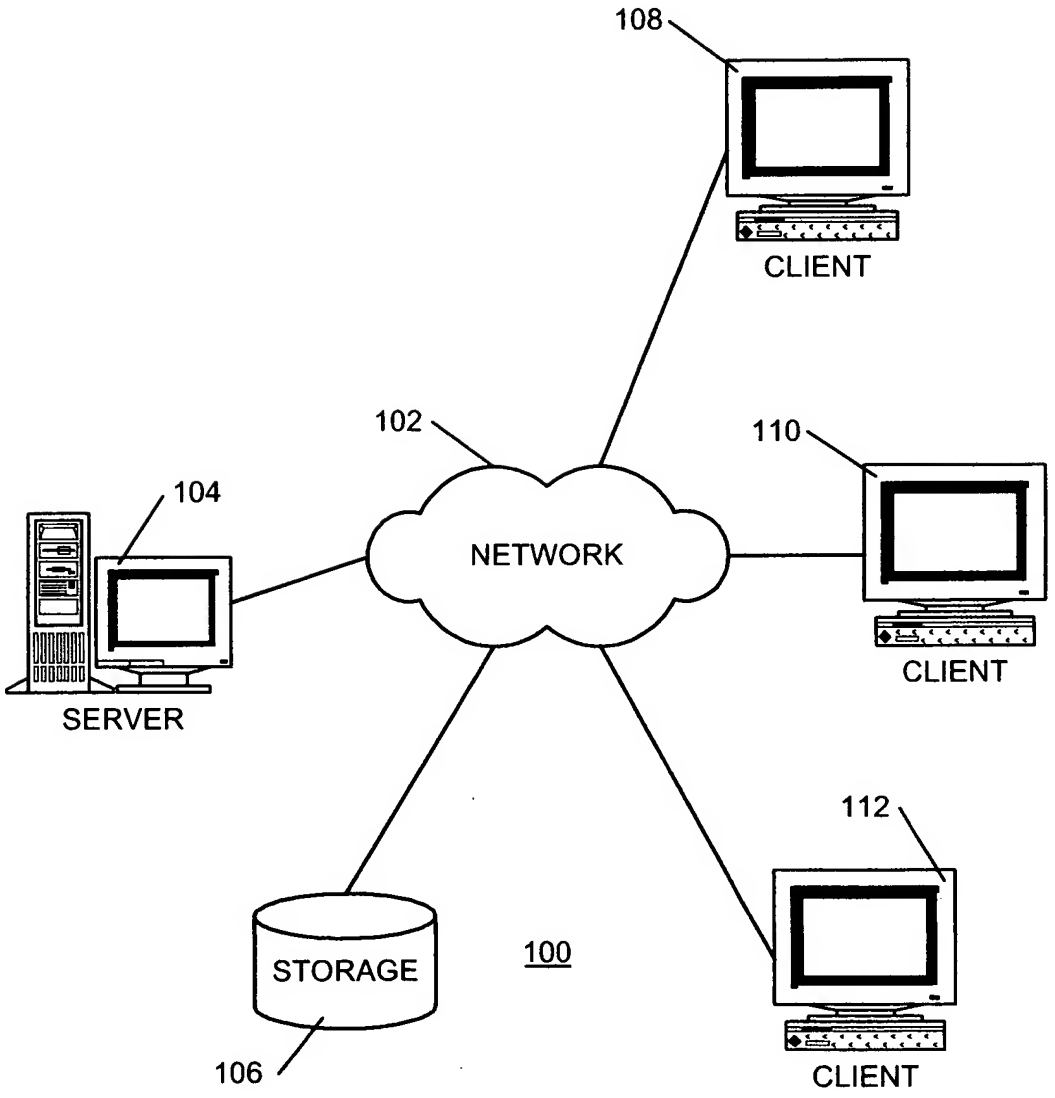

FIG. 1

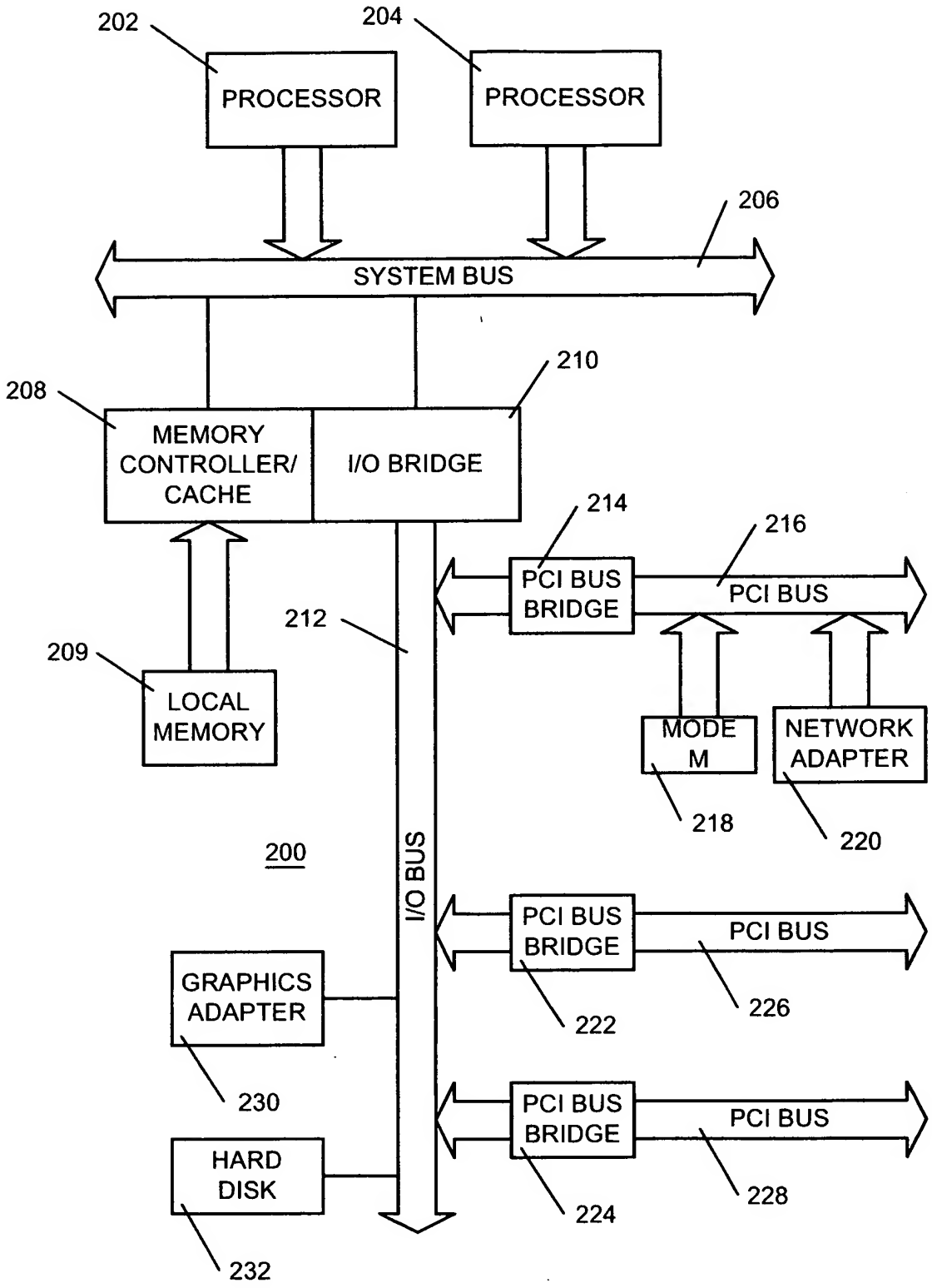
202
204
PROCESSOR
PROCESSOR
206
SYSTEM BUS
208
210
MEMORY CONTROLLER/ CACHE
I/O BRIDGE
214
216
PCI BUS BRIDGE
PCI BUS
212
209
LOCAL MEMORY
MODE M
NETWORK ADAPTER
218
220
200
I/O BUS
PCI BUS BRIDGE
PCI BUS
GRAPHICS ADAPTER
222
226
230
PCI BUS BRIDGE
PCI BUS
HARD DISK
224
228
232

FIG. 2

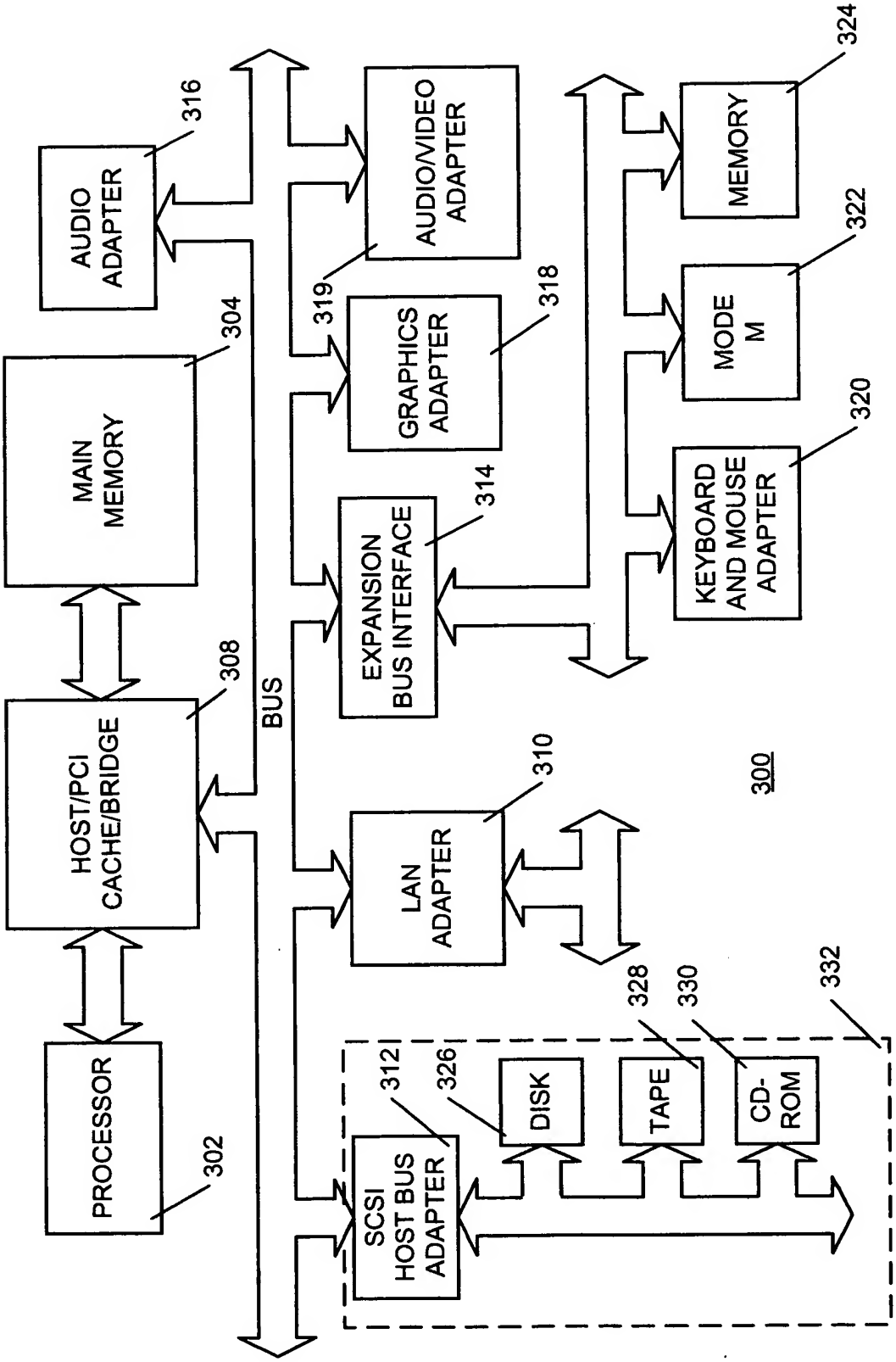
PROCESSOR
302
HOST/PCI CACHE/BRIDGE
308
MAIN MEMORY
304
AUDIO ADAPTER
316
BUS
AUDIO/VIDEO ADAPTER
319
GRAPHICS ADAPTER
318
EXPANSION BUS INTERFACE
314
LAN ADAPTER
310
SCSI HOST BUS ADAPTER
312
DISK
326
TAPE
328
CD-ROM
330
332
300
MEMORY
324
MODE M
322
KEYBOARD AND MOUSE ADAPTER
320

FIG. 3

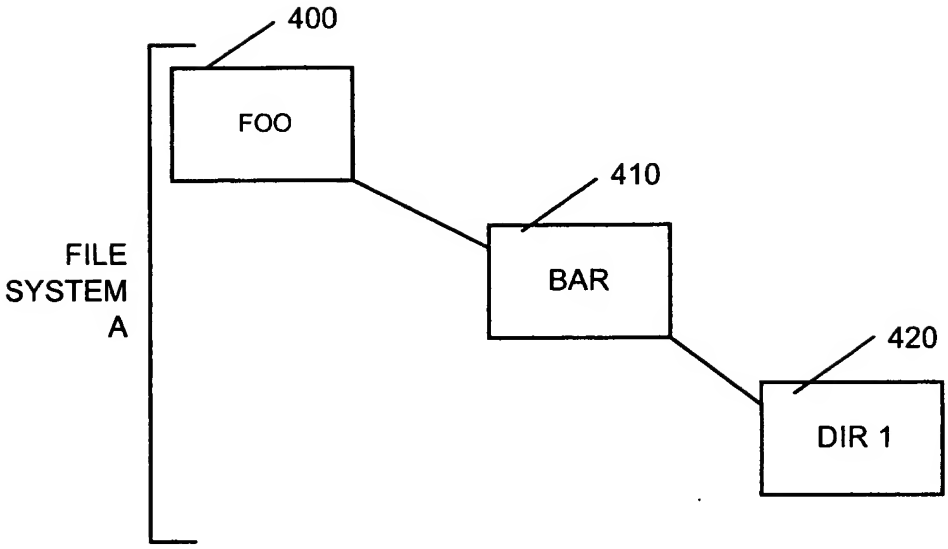
400
FOO
410
BAR
420
DIR 1
FILE
SYSTEM
A

FIG. 4

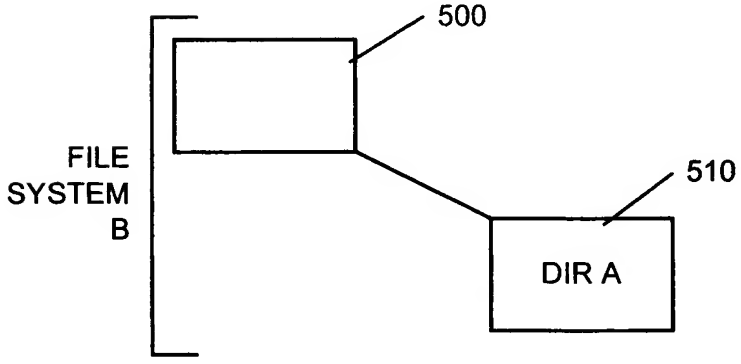
500
510
DIR A
FILE
SYSTEM
B

FIG. 5

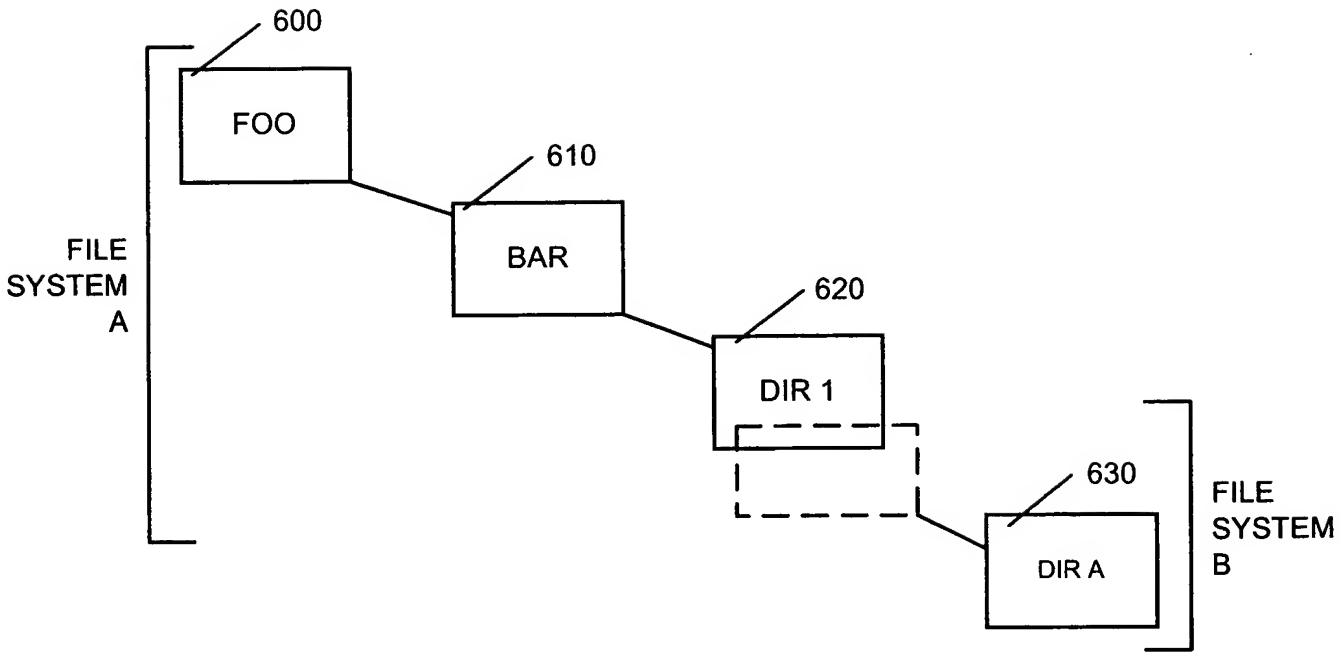
600
FOO
610
BAR
620
DIR 1
630
DIR A
FILE SYSTEM A
FILE SYSTEM B

FIG. 6

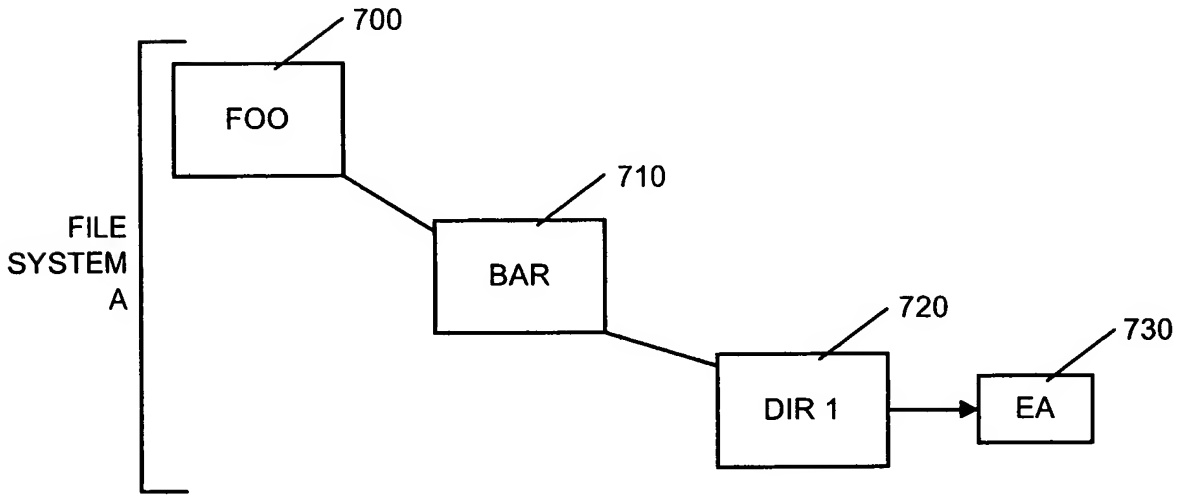
700
FOO
710
BAR
720
DIR 1
730
EA
FILE SYSTEM A

FIG. 7

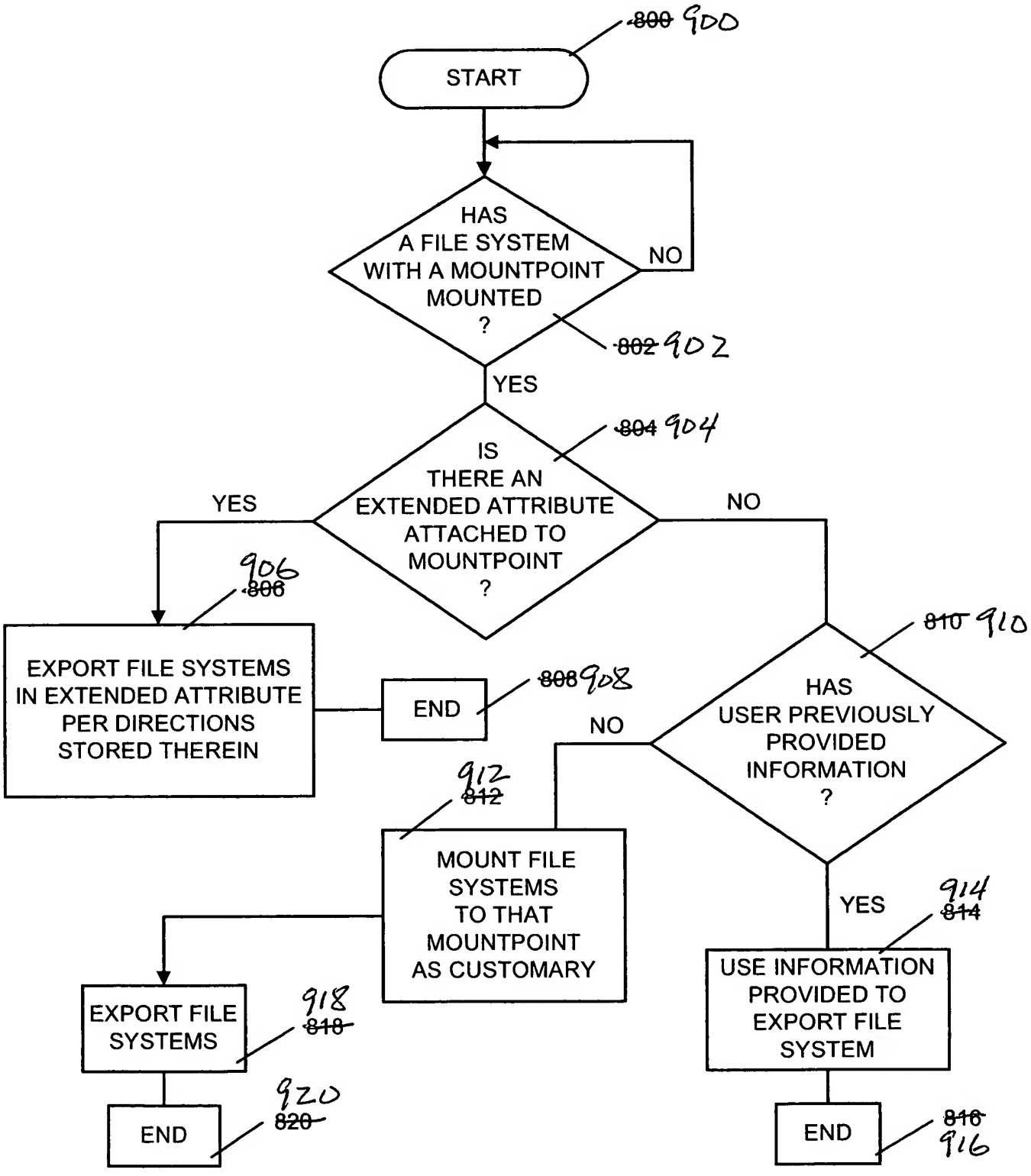
~~800~~ 900
START
HAS A FILE SYSTEM WITH A MOUNTPOINT MOUNTED ?
NO
~~802~~ 902
YES
~~804~~ 904
IS THERE AN EXTENDED ATTRIBUTE ATTACHED TO MOUNTPOINT ?
YES
NO
906
~~806~~
EXPORT FILE SYSTEMS IN EXTENDED ATTRIBUTE PER DIRECTIONS STORED THEREIN
END
~~808~~ 908
~~810~~ 910
HAS USER PREVIOUSLY PROVIDED INFORMATION ?
NO
912
~~812~~
MOUNT FILE SYSTEMS TO THAT MOUNTPOINT AS CUSTOMARY
YES
914
~~814~~
USE INFORMATION PROVIDED TO EXPORT FILE SYSTEM
EXPORT FILE SYSTEMS
918
~~818~~
END
920
~~820~~
END
~~816~~ 916

FIG. ~~8~~ 9